

第五次上机作业（指针、结构体）求解思路

说明：

- 作业可用多种方法求解，可使用本文档思路，也可按自己的想法设计程序；
- 若程序中有逻辑问题（运行结果与预想不同），可使用“单步调试”查找具体出现逻辑问题的语句，详见“超星学习通”1.2节视频最后部分。

指针编程题

1. 字符串更新：利用指针编写函数update()，将长度为10的字符型数组，从下标为6的元素开始全部设置为'#'，并在主函数输出更新后的字符串。

提示：

(1) main函数中，定义长度为10的字符数组，从键盘输入字符串，并以数组名为参数传递数组首地址（地址传递），输出更新后的字符串；

(2) update函数中，形参应设置为字符指针或字符数组形式，利用循环实现将下标为6到下标为8的数组元素赋值为'#'。循环之后，对下标为9的元素赋值'\0'结束字符串。由于函数采用地址传递，因此主函数内定义的字符数组中元素随之变化。

更新后字符串的输出结果分两种情况：

①若输入字符串的字符数≤5时，原字符串结束符'\0'出现在下标为6的元素前，因此填入的'#'只是赋值到数组中但无法显示；②若输入的字符串的字符数>5时，填入的'#'将覆盖原结束符'\0'。



2. 完善函数value()：找出10个元素中最大值和最小值，并用语言描述value函数的执行过程，及如何将调用结果返回main函数。

```
#include<stdio.h>
```

```
void value(int a[],int *p1,int *p2, int n)
```

```
{
```

补充完整

```
}
```

```
void main()
```

```
{ int max=0,min=0,*mx,*mn;
```

```
int arr[]={1,2,3,-5,0,7,15,4,-8,10};
```

```
mx=&max, mn=&min;
```

```
value(arr,mx,mn,10);
```

```
printf("%d, %d\n", max, min);
```

```
}
```

提示：

按照经典算法“求最值”实现即可，由于函数形参p1和p2均为指针形式（地址传递），因此在函数中能够通过*p1和*p2改变主函数中max和min的值。

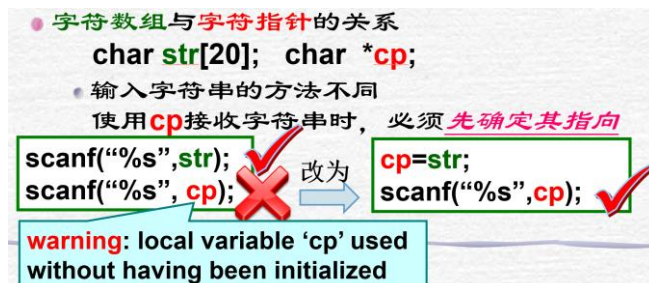
3. 字符串排序：在主函数中输入10个不等长的字符串放入二维字符数组中，编写函数**sort()**利用指针数组对其排序，在主函数中输出排好序的字符串。

函数原型为：void sort(char *s []);

提示：

(1) main函数中，需定义二维字符数组（行数=10）用于存放10个字符串，定义字符指针数组（长度=10）作为实参（以确保与sort函数的形参一致）；

必须定义字符数组的原因是不能用字符指针直接接收字符串：



main函数中先利用循环从键盘接收10个字符串赋值给二维字符数组，再用一个循环使得字符指针数组中的各个指针指向二维数组中的各行。

以字符指针数组名为实参调用sort函数，之后输出排好序的字符串；

(2) sort函数中，利用课件上“交换地址的简单选择排序”即可实现多个字符串排序。

4. 学生成绩统计：有一个班级6个学生5门课（每门课程及平均分均为整数），分别以数组和指针为形参编写函数实现下列功能：

(1) 函数 int average() 求第 n 个指定课目的平均分；

(2) 函数 void findfail() 找出有 2 门以上（含 2 门）课程不及格的学生，在主函数输出其学号和全部课程成绩及平均成绩；

(3) 函数 void findgood() 找出平均成绩在 90 分以上或全部课程成绩在 85 分以上的学生，在主函数输出其学号和全部课程成绩及平均成绩。

提示：设置符号常量 N 为 6（学生数），M 为 5（课程数）

(1) main函数中，定义 int s[N][M+3]，多出的 3 列分别存放“学号”、“平均分”、“成绩标志”（2 门及 2 门以上不及格标为 'N'，平均成绩在 90 分以上或全部课程成绩在 85 分标为 'Y'，由于 int 型和 char 型具有“互通性”，因此能够将 'N'、'Y' 存放在 int 型数组中）；

利用循环输入 N 个学生的学号、M 门成绩；

输入要计算平均分的课程序号 n，以 s 和 n 为参数调用 average()，并输出相应结果；

以 s 为参数调用 findfail()，并利用循环输出数组 s 中所有“成绩标志”为 'N' 的学生的学号、全部课程成绩及平均成绩；

以 s 为参数调用 findgood()，并利用循环输出数组 s 中所有“成绩标志”为 'Y' 的学生的学号、全部课程成绩及平均成绩；

(2) average函数中，形参为二维数组 s 和课程序号 n，利用循环累加计算 N 个学生第 n 门课程的总分（若设置“学号”的列下标为 0，则第 n 门课程的列下标为 n）。返回总分/N，平均成绩为整型；

(3) findfail函数中，形参为二维数组 s，用循环对每名学生进行如下操作：计数 count 初始设为 0，用于统计 <60 分的成绩数量；

总分 sum 初始设为 0，用于累加得到总分，进而计算每名学生的平均分；

利用循环累加计算第 i 个学生的总分，在循环中同时判断第 i 个学生第 j 门课

成绩是否 <60 ，为真则 $\text{count}++$;

将第 i 个学生的平均分存入该学生“平均分”对应的数组元素中;

若 $\text{count} \geq 2$ ，则将‘N’赋值给该学生“成绩标志”对应的数组元素中;

由于 findfail 函数的形参为二维数组名 s ，因此通过地址传递 main 函数中二维数组中的值随之改变;

(4) findgood 函数与 findfail 函数类似。

5. 完善下面的程序：从键盘输入 n ($n < 10$) 个不重复的字符串，找出其中最小的字符串，将其与第一个字符串对换，其他字符串不动。在方框中填写语句，将程序补充完整，并对完善后的程序加上适当的注释。

```
#include <stdio.h>
#include <string.h>
#define N 10
#define M 80
void input( )
{
    int n=0;
    printf("Input %d strings:\n",len);
    while(n<len)
        gets( );
}
void min_str( char (*str)[M], int len )
{
    char *min, (*p)[M], temp[M];
    min=str[0];
    for( p=str+1; p<str+len; p++)
        if(  )
            min=*p;
    strcpy(temp,min);
    ;
    strcpy(*str, temp);
}

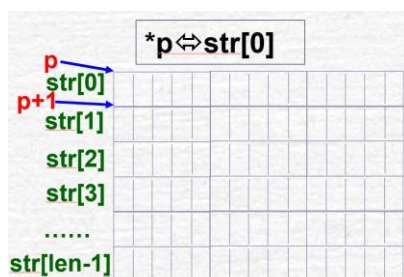
void output(char s[ ][M], int len)
{
    int n;
    printf("\nThe result is:\n");
    for(n=0; n<len; n++)
        printf( "%s\n",s[n]);
}

void main()
{
    char str[N][M];
    int n;
    printf("Input n (<10):");
    scanf("%d", &n);
    getchar();
    input( str, n);
    min_str(  );
    output( str, n );
}
```

提示：

(1) input 函数中，第一个形参可设置为二维字符数组或一维数组指针变量，与 main 函数调用语句中的第一个实参 str 一致，第二个形参为整型。函数中循环体功能为对二维数组中第 n 个字符串赋值，赋值后 n 需自加；

(2) min_str 函数中，可用 p 指向二维数组中的各行（即各字符串），因此可用 $*p$ 表示当前字符串的首地址。



$\text{min}=\text{str}[0]$;设置了 min 的初始指向，使得 min 初始指向行下标为0的字符串，与查找最小值算法一致；

for 循环的功能是利用 p 指向字符串 $\text{str}[1]$ （地址为 $\text{str}+1$ ）到 $\text{str}[\text{len}-1]$ （地址为 $\text{str}+\text{len}-1$ ），循环体中每次将 min 指向的字符串与 $*p$ 指向的字符串进行是否大于的比较（利用 strcmp 函数），使得 min 始终指向最小字符串；

之后利用辅助字符数组temp、3条语句实现交换；

(3) main函数中，调用min_str函数需要以二维字符数组名str及字符串个数n为实参，与min_str函数的形参一致。

结构体编程题：

6. 结构体程序设计：考研初试成绩包括外语、政治、数学和专业课四科成绩（其中，外语和政治满分 100 分，数学和专业课满分 150 分）。某校复试分数要求是同时满足：外语、政治不低于 45 分；数学和专业课不低于 75 分；总分不低于 320 分。

编程实现 5 名考生考研成绩统计功能。

- ① 定义结构体类型，每个考生信息包括：考号（1~5）、姓名、各科成绩、初试总分和是否满足复试要求标记；
- ② 编写函数 Cal_Mark()，计算每个考生的初试总分，并将不满足复试要求的考生标记为'F'，满足复试要求的考生标记为'P'；
- ③ 编写函数 SelectionSort()，用简单选择排序法按初试总分从高到低排序；
- ④ 编写主函数，在主函数中从键盘输入所有考生的考号、姓名和四科成绩，通过参数传递调用上述两个函数，最后在主函数中按初试总分从高到低顺序输出满足复试要求的考生信息组成的复试名单。

提示：

- (1) 在所有函数外定义结构体类型，则所有函数均可使用；
- (2) SelectionSort 函数中进行排序，注意交换的是每个考生的所有信息，因此辅助变量为结构体类型。

思考题（选做）

7. 统计歌词中的单词个数，并找出最短和最长的单词。设计函数实现：

- ① 函数 int gecifun(char *geci, char *strmax, char *strmin)——实现求 geci 中的单词个数并返回，strmax 和 strmin 中分别存放最长和最短单词（不计标点）；
- ② 主函数——调用函数 gecifun 并输出。
- ③ 程序输出：歌词中共有 XX 个单词，歌词中最长的单词为 Brother，最短的单词为 I。

歌词：Daddy finger, Daddy finger, where are you? Here I am. Here I am.
How do you do? Mammy finger, Mammy finger, where are you? Here I am.
Here I am. How do you do? Brother finger, Brother finger, where are you? Here I am. Here I am. How do you do?

提示：

(1) main 函数中，定义字符数组 geci，初始值为字符串常量，并定义字符数组 strmax、strmin 用于存放最长、最短单词；

(2) gecifun 函数中，形参的 3 个字符型指针分别接收 main 函数中定义 3 个数组的首地址；

定义字符数组 strqu 存放取出的每个单词，定义 int count 统计单词个数，定义 int maxlen（初始值为 0）和 int minlen（初始值设为一个很大的整数）表示当前的最长、最短单词长度；

从字符数组中取出单个单词方法请见“第五章数组”PPT 课件中例题。

字符数组和字符串

例 取字符串中的第一个单词(单词分隔符为 ,. ─)。

- ①题目要求取出歌词中的所有单词，因此需加外层循环 `while(*geci!='\0');`
 - ②用变量 `j` 表示 `strqu` 中下标时，取每个单词前均需设置 `j=0`，即每个单词都从 `strqu` 下标为 0 的位置开始存放字符；
 - ③利用循环取单词，单词分隔符包括 `' '`、`' '`、`'.'` 和 `'?'`；
 - ④循环后需为 `strqu` 添加结束符 `'\0'`；
 - ⑤当 `j>0` 时表示取出了一个单词，`count` 自加，用取出单词 `strqu` 的长度(`strlen` 函数)与当前找到的单词最大长度 `maxlen` 和单词最小长度 `minlen` 进行比较，根据比较结果更新 `maxlen` 和 `minlen` 的值并为 `strmax` 和 `strmin` 赋值。
- 判断 `j>0` 的原因是：若③的循环只取到了某一分隔符，则④使得 `strqu` 中只保存了 `'\0'`，即这次循环并未取出单词，只是“跳过”了分隔符。
- (3) 注意：对字符串操作需要用相关函数，如 `strlen` 函数、`strcpy` 函数，并包含头文件 `string.h`。